

Durham Research Online

Deposited in DRO:

03 October 2016

Version of attached file:

Accepted Version

Peer-review status of attached file:

Peer-reviewed

Citation for published item:

Coelho, A.A. and Evans, J.S.O. and Lewis, J.W. (2016) 'Averaging the intensity of many-layered structures for accurate stacking-fault analysis using Rietveld refinement.', *Journal of applied crystallography.*, 49 (5). pp. 1740-1749.

Further information on publisher's website:

<http://dx.doi.org/10.1107/S1600576716013066>

Publisher's copyright statement:

Additional information:

Use policy

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a [link](#) is made to the metadata record in DRO
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Please consult the [full DRO policy](#) for further details.

Averaging the intensity of many layered structures for accurate stacking fault analysis using Rietveld Refinement

Alan A. Coelho^(a), John S. O. Evans^(b), James W. Lewis^(b)

^(a) 72 Cedar Street, Wynnum, 4178, Brisbane, Australia, e-mail: AlanCoelho@bigpond.com

^(b) Department of Chemistry, University Science Site, Durham University, South Road, Durham, DH1 3LE, UK

Keywords: Stacking faults, Rietveld refinement, crystallography, TOPAS software

Abstract

Many technologically important synthetic and natural materials display stacking faults which lead to complex peak broadenings, asymmetries and shifts in their powder diffraction patterns. The patterns can be described using an enlarged unit cell (called a supercell) containing an explicit description of the layers. Since the supercell can contain hundreds of thousands of atoms with hundreds of thousands of hkl reflections, a Rietveld approach has been too computationally demanding for all but the simplest systems. We describe the implementation of the speed-ups necessary to allow Rietveld refinement in the computer program TOPAS Version 6 [1]. Techniques implemented include: a peaks-buffer that allows hundreds of thousands of hkl -dependent peak shapes to be automatically approximated by a few hundred peaks; an averaging process for hundreds of large supercells with minimum impact on computational time; a smoothing technique that allows for the use of small supercells which approximate supercells 10 to 20 times larger; and efficient algorithms for stacking sequence generation. The result is Rietveld refinement of supercells operating at speeds 1000s of times faster than that of traditional Rietveld refinements. This allows quantitative and simultaneous analysis of structure and microstructure in complex stacking-faulted samples.

[1] Bruker AXS (2015). TOPAS, V6.0. *Bruker AXS, Karlsruhe, Germany*

1. Introduction

Quantitative Rietveld analysis of layered structures exhibiting stacking vector faulting is becoming more common as computers become more powerful and the analytical software more accommodating. The supercell approach (Ufer *et al.*, 2004; Wang *et al.*, 2012; Lutterotti *et al.*, 2010) uses multiple layers in an enlarged unit cell in order to simulate faulted layered structures; it was used alongside recursive DIFFaX simulations (Treacy *et al.*, 1991) using TOPAS Version 5 by Bette *et al.* (2015). In that work many stacking sequences relating to the structure of NiCl(OH) were trialled with each run taking days of computing time. This work addresses the time consuming computational aspects of the supercell approach and is an extension of the work by Ainsworth *et al.* (2016). Bottle necks associated with unit cells comprising thousands of atoms resulting in hundreds of thousands of reflections are tackled. In particular the three hour 1000 repeat Rietveld refinements performed by Ainsworth *et al.* (2016) for averaging of supercells can now be performed in under a minute.

Rietveld refinement (Rietveld, 1969; Young, 1993) can be broken down into a number of steps; i) calculation of structure factors, ii) calculating peaks with intensities obtained from the structure factors, iii) forming a diffraction pattern using the calculated peaks and iv) calculating derivatives for parameters that are refined. We address the challenges associated with each of these steps then exemplify the computational speed gains possible.

2. Calculating structure factors

The structure factor for a particular hkl of a structure that can be described in terms of stacked layers is the complex quantity:

$$F_{str,hkl} = \sum_L \sum_v \sum_a O_{L,a} T_{L,a} \left(f_{0,L,a} + f'_{L,a} + i f''_{L,a} \right) e^{2\pi i (\mathbf{h} \cdot (\mathbf{r}_{L,a} + \mathbf{v}_{L,v}))} \quad (1)$$

Where the subscript L corresponds to the different layer types, v to stacking vectors of layer L and a to the atoms in layer L . $\mathbf{r}_{L,a}$ corresponds to the atom vector position in fractional coordinates, $T_{L,a}$ the temperature factor, $O_{L,a}$ the site occupancy, $f_{o,L,a}$ the atomic scattering factor, $f'_{L,a}$ and $f''_{L,a}$ the anomalous dispersion coefficients, \mathbf{h} the hkl vector $\{h, k, l\}$ and $\mathbf{v}_{L,v}$ the stacking vector v of each layer in fractional coordinates. The stacking vector part of Eq. 1 can be taken out of the innermost summation, as shown in Eq. 2, allowing the stacking vectors summation to be performed independently of the layers summation.

$$F_{str,hkl} = \sum_L \left(\sum_v e^{2\pi i (\mathbf{h} \cdot \mathbf{v}_{L,v})} \right) Layer_{L,hkl} \quad (2)$$

$$\text{where } Layer_{L,hkl} = \sum_a Atom_{L,a}$$

$$\text{and } Atom_{L,a} = O_{L,a} T_{L,a} \left(f_{o,L,a} + f'_{L,a} + i f''_{L,a} \right) e^{2\pi i (\mathbf{h} \cdot \mathbf{r}_{L,a})}$$

In many cases the number of unique x and y coordinates N_{uxy} amongst the stacking vectors $\{\mathbf{v}_{L,v,x}, \mathbf{v}_{L,v,y}\}$ is small and Eq. 2 can be rearranged to reflect this as shown in Eq. 3 where the summation over vxy corresponds to stacking vectors with unique $\{\mathbf{v}_{L,v,x}, \mathbf{v}_{L,v,y}\}$ and the summation over u corresponds to $\mathbf{v}_{L,v,z}$ values that all have the same $\{\mathbf{v}_{L,v,x}, \mathbf{v}_{L,v,y}\}$. The resulting $V_{L,hkl}$ is a complex quantity derived from all the stacking vectors as applied to layer L . Performing the summation of Eq. 2 using Eq. 3 can speed up the calculation of $F_{str,hkl}$ by a factor of 15 to 20.

$$F_{str,hkl} = \sum_L V_{L,hkl} Layer_{L,hkl} \quad (3)$$

$$\text{where } V_{L,hkl} = \sum_{vxy} e^{2\pi i (h v_{x,vxy} + k v_{y,vxy})} \sum_u e^{2\pi i l v_{z,vxy,u}}$$

For analyses where $\{\mathbf{v}_{L,v,x}, \mathbf{v}_{L,v,y}\}$ vectors are randomly offset, as in the work of Bette *et al.* (2015), then the speed gain using Eq. 3 is not possible.

3. Obtaining an average intensity from many similarly faulted crystals

To more accurately represent the probabilistic nature of stacking faults using supercells of finite size an average intensity from many crystals (called structures from here on) can be used. One could perform this averaging from N_{str} separate pattern calculations, where N_{str} corresponds to the number of structures used, which would require an N_{str} fold increase in computational time. The process can, however, be performed by averaging the intensities as described in Eq. 4 and then calculating one pattern instead of N_{str} patterns.

$$I_{avg,hkl} = (1/N_{str}) \sum_{str} |F_{str,hkl}|^2 \quad (4)$$

For structures comprising layer types of various thicknesses, the thickness of each structure will vary due to faulting. A finite and small N_v would then give N_{str} slightly different lattice parameters. Efficient use of Eq. 4, however, requires the use of a single set of lattice parameters for all structures; for stacking along the c -axis this means using an average c lattice parameter c_{avg} . This allows the use of the same set of hkl s and the same calculated peak shapes in the Rietveld calculations, and means that summing the calculated peaks onto the calculated pattern is performed once and not N_{str} times. The neglected smoothing due to differing lattice parameter can be approximated in a smoothing function described below.

The speed at which $I_{avg,hkl}$ can be calculated for 100 structures in the generation of a typical powder pattern is only two to three times slower than I_{hkl} for a single structure; this is demonstrated below. In the theoretical recursive method of Treacy *et al.*, (1991) the intensity for a particular faulting probability is treated statistically. The present work achieves a similar effect numerically using $I_{avg,hkl}$.

From equations 3 and 4 the computational effort, $O(I_{avg,hkl})$, required to calculate $I_{avg,hkl}$ scales by Eq. 5.

$$O(I_{avg,hkl}) = N_{str} (N_v l_{max} + N_{uxy} N_{hkl}) \quad (5)$$

l_{max} corresponds to the maximum l value encountered in the calculation of $V_{L,hkl}$ of Eq. 3.

4. Generating Stacking Sequences

In the generation of the stacking sequences from a particular stacking probability matrix it is advantageous to have $I_{avg,hkl}$ changing as little as possible. This allows the goodness of fit of multiple Rietveld refinements to be compared as a function of the parameters of the stacking probability matrix. Indeed the change in the goodness of fit can be determined as a function of the probability parameters and optimization performed in order to determine best-fit values.

Stacking sequences are generated from a stacking probability matrix of the type used in the DIFFaX package (Treacy, 1991, see for example Figure 9 of Ainsworth *et al.* 2016). In an initial step the probability matrix is used to generate $N_{avg} N_{str} N_v$ layers and from this prediction run the number of each layer i to j transition is determined and placed in a summation matrix \mathbf{S} which is then divided by N_{avg} to give \mathbf{S}_{pred} . N_{avg} is typically set to 100. Stacking sequences for each structure are then generated according to the steps shown in Table 1. This procedure ensures that the generated summation matrix \mathbf{S}_{gen} closely matches \mathbf{S}_{pred} which in turn matches the overall faulting expressed in the transition matrix. The overall time for generating sequences is small compared to the Rietveld calculations. On termination of the generation process an average c lattice parameter c_{avg} from the N_{str} structures is determined and subsequently used in the calculation of peak positions.

5. Smoothing ripples and correcting for differing layer thicknesses

For layer types with differing thicknesses the generation scheme in Table 1 will produce supercells of slightly varying thicknesses (varying c lattice parameters) resulting in smearing of the peaks. This smearing is not described by $I_{avg,hkl}$ (Section 3) due to the use of a finite and single c_{avg} lattice parameter in the calculation of peak position. In addition, stacking

faulted diffraction patterns calculated from small N_v supercells will contain ripples and oscillations from individual hkl reflections when the instrument, emission profile and sample aberrations are small. Using $I_{avg,hkl}$ reduces ripples caused by poor faulting statistics, but does not eliminate the ripples caused by small N_v . Ainsworth *et al.* (2016) therefore used a large supercell, $N_v=5000$, to minimise both effects and generate a smooth pattern. Being able to reduce N_v whilst i) reducing ripples and ii) describing smearing due to the use of a single c lattice parameter without degrading the faulted pattern would be hugely beneficial; the number of reflections would be greatly reduced resulting in much less computer memory allocation and far greater computational speed.

As N_v increases the number of peaks increase; in particular stacking along the crystallographic c axis sees peaks inserted between the (h_1, k_1, l_1) and (h_1, k_1, l_1+1) peaks where (h_1, k_1, l_1) are the Miller indices when $N_v=1$. When the distance in 2θ from (h, k, l) to $(h, k, l+1)$ is less than the step size of the data for all peaks then the resulting diffraction pattern is smooth. For small N_v and sharp intrinsic peaks it is possible to produce a smooth pattern comparable to one created with a large N_v by inserting peaks, irrespective of N_v , from $2\theta_{h,k,l}$ to $2\theta_{h,k,l+1}$ with intensities ranging from $F_{h,k,l}^2$ to $F_{h,k,l+1}^2$. This introduction of peaks can be approximated by convoluting a right trapezoid of size $d2\theta/dl=(2\theta_{h,k,l+1}-2\theta_{h,k,l})$ into each hkl reflection with intensities at its ends corresponding to $F_{h,k,l}^2$ and $F_{h,k,l+1}^2$. Similarly a right trapezoid of size $(2\theta_{h,k,l}-2\theta_{h,k,l-1})$ can be convoluted to insert peaks from $2\theta_{h,k,l-1}$ to $2\theta_{h,k,l}$. This twin right trapezoidal function can be approximated by a Gaussian centered at $2\theta_{h,k,l}$ and then shifted by the centroid shift of the twin trapezoid as shown in Eq. 6a.

Not all peaks should be broadened by $d2\theta/dl$. For example, if all layers have the same height, peaks with indices $(0,0,nN_v)$ where n is an integer won't be broadened; nor would other categories of (h,k,nN_v) peaks depending on the faulting mechanism. When going from a no-

faulting case to a faulting case, lattice points with unchanging perpendicular distances to a particular plane will have their corresponding peak remain sharp. One way of testing whether (h,k,l) should be broadened is by looking at the structure factors. If $F_{h,k,l-1}^2$ and $F_{h,k,l+1}^2$ are near zero and $F_{h,k,l}^2$ is large, then little to no broadening is expected. On the other hand if $F_{h,k,l-1}^2$, $F_{h,k,l}^2$ and $F_{h,k,l+1}^2$ are all of a similar intensity then $d\theta/dl$ broadening is expected. If we assume the broadening is dependent on $1 - F_{h,k,l}^2 / (F_{h,k,l-1}^2 + F_{h,k,l}^2 + F_{h,k,l+1}^2)$ we arrive at the broadening term $\Delta_{h,k,l}$ shown in Eq. 6b.

$$Shift_{h,k,l} = \left(\frac{(2\theta_{h,k,l-1} - 2\theta_{h,k,l})F_{h,k,l-1}^2 + (2\theta_{h,k,l+1} - 2\theta_{h,k,l})F_{h,k,l+1}^2}{F_{h,k,l-1}^2 + F_{h,k,l}^2 + F_{h,k,l+1}^2} \right) \quad (6a)$$

$$\Delta_{h,k,l} = \tau d2\theta/dl \left(\frac{F_{h,k,l-1}^2 + F_{h,k,l+1}^2}{F_{h,k,l-1}^2 + F_{h,k,l}^2 + F_{h,k,l+1}^2} \right) \quad (6b)$$

τ is a parameter that is varied in order to obtain the best fit during refinement. $\Delta_{h,k,l}$ is small when $F_{h,k,l-1}^2$ and $F_{h,k,l+1}^2$ are small and $F_{h,k,l}^2$ is large; it is large when $F_{h,k,l-1}^2$, $F_{h,k,l}^2$ and $F_{h,k,l+1}^2$ are of similar intensity. $\Delta_{h,k,l}$ corresponds to the FWHM in 2θ of a Gaussian function convoluted into each peak and $Shift_{h,k,l}$ is the shift applied to each convolution. Eq. 6 represents a smoothing function that allows small supercells to approximate much larger supercells. In addition to reducing ripples due to a small N_v it also approximates the smearing not seen when there are layer types of varying thicknesses and a single set of lattice parameters used.

This approach produces calculated patterns comparable to supercells 10 to 20 times larger. Specifically for two cases a and b with $N_{v,a}$ and $N_{v,b}$ layers, the speed gain in terms of calculating $I_{avg,hkl}$ using Eq. 5 is shown in Eq. 7.

$$\text{Gain} = O_a(I_{\text{avg},hkl})/O_b(I_{\text{avg},hkl}) = \quad (7)$$

$$N_{\text{str},a} (N_{v,a} l_{\text{max},a} + N_{\text{uxy},a} N_{hkl,a}) / (N_{\text{str},b} (N_{v,b} l_{\text{max},b} + N_{\text{uxy},b} N_{hkl,b}))$$

where $O(I_{\text{avg},hkl})$ is the computational effort to calculate $I_{\text{avg},hkl}$. For cases a and b to have similar faulting statistics then $N_{\text{str},a} N_{v,a} = N_{\text{str},b} N_{v,b}$. Noting also that $l_{\text{max},a}/l_{\text{max},b} = N_{v,a}/N_{v,b}$ and $N_{hkl,a}/N_{hkl,b} = N_{v,a}/N_{v,b}$, after substitution into Eq. 7 we get the speed gain shown in Eq. 8.

$$\text{Gain} = (N_{v,a} l_{\text{max},b} + N_{\text{uxy},a} N_{hkl,a}) / (N_{v,b} l_{\text{max},b} + N_{\text{uxy},b} N_{hkl,b}) \quad (8)$$

For structures with small N_{uxy} Eq. 8 reduces to $\sim N_{v,a}/N_{v,b}$. If $v_{L,v,x}$ and $v_{L,v,y}$ were random values then $N_{\text{uxy},a}$ and $N_{\text{uxy},b}$ becomes $N_{v,a}$ and $N_{v,b}$ for systems a and b respectively. The speed gain would then be exactly $N_{v,a}/N_{v,b}$.

6. Calculating derivatives of structural parameters

Derivatives of $I_{\text{avg},hkl}$ with respect to stacking vector coordinates, atomic coordinates, occupancies, atomic scattering factors, anomalous dispersion coefficients or temperature factors are all necessary in order to refine on the corresponding parameters in a non-linear least squares sense. Excluding the stacking vector coordinates, these derivatives can all be calculated using a layer dependent $N_{\text{str},x} N_{hkl}$ matrix calculated each non-linear least-squares iteration. Consider the case of calculating the derivative with respect to the fractional atomic coordinate $r_{L,a,x}$ for atom a of layer L . From Eq. 3 we have:

$$\partial I_{\text{avg},hkl} / \partial r_{L,a,x} = 2 F_{\text{str},hkl} V_{L,hkl} \text{Atom}_{L,a} 2\pi i h \quad (9)$$

In each derivative the quantity $2 F_{\text{str},hkl} V_{L,hkl}$ is used. Both $F_{\text{str},hkl}$ and $V_{L,hkl}$ are used during the calculation of $I_{\text{avg},hkl}$ and hence do not need recalculating for the derivatives. Thus once $I_{\text{avg},hkl}$ is calculated, determining the derivatives for a structural parameter for a multi-structure refinement takes the same computational effort as for a single-structure refinement.

For the stacking vector coordinates, the derivative from Eq. 3 for the x coordinate of stacking vector v becomes:

$$\partial I_{avg,hkl} / \partial v_{L,x} = 2 F_{str,hkl} Layer_{L,hkl} e^{2\pi i (\mathbf{h} \cdot \mathbf{v}_L)} 2\pi i h \quad (10)$$

Again, both $F_{str,hkl}$ and $Layer_{L,hkl}$ are calculated during the calculation of $I_{avg,hkl}$ and hence do not need recalculation for stacking vector coordinate derivatives. Even though these derivatives can be calculated in a fast manner, coordinates of the stacking vectors are typically not refined independently as the quality of the data does not support using 1000s of such parameters. Instead thicknesses of layer-types might be refined resulting in fewer than a dozen or so independent parameters.

7. Computational effort as compared to traditional Rietveld refinement

Rietveld refinement without the use of stacking vectors requires the explicit definition of atoms in each stacked layer. We can consider the computational effort using a traditional approach (e.g. TOPAS Version 4, Bruker (2009)) for averaging $N_{str}=100$ structures each with $N_v=960$ layers using the $\text{La}_2\text{O}_2\text{Cu}_{0.667}\text{Cd}_{0.667}\text{Se}_2$ example investigated by Ainsworth *et al.* (2016). Each structure has $N_{hkl}=54881$ *hkl*s, six layer types $N_L=6$ with four layers having seven atoms per layer and two layers having eight atoms per layer. Each peak shape generated requires $N_{ops_per_pk} \approx 2000$ operations and each summation of the peak shape to the calculated pattern requires $N_{pk_shape_sum} \approx 150$ operations. The value of $N_{ops_per_pk} \sim 2000$ corresponds to a pseudo-Voigt peak shape convoluted with one emission profile line with the instrument aberrations of axial divergence and a receiving slit width (Cheary & Coelho, 1992). The computational effort required to calculate $I_{avg,hkl}$, $O(I_{avg,hkl})$, and to calculate the peaks, $O(Peaks)$, approximately scales according to Eq. 11 with all numbers originating from the program itself.

$$O(I_{avg,hkl}) = N_{str} N_{hkl} \left(\sum_L N_{L,atoms} \right) N_v / N_L \quad (11)$$

$$= 100 * 54881 * 44 * 960 / 6 = 3.86 \times 10^{10}$$

$$O(Peaks) = N_{hkl} \left(N_{ops_per_pk} + N_{pk_shape_sum} \right)$$

$$= 54481 * (2000 + 150) = 1.17 \times 10^8$$

The same calculation in TOPAS Version 6 with the stacking implementations discussed above reduces to the scaling shown in Eq. 12.

$$O(I_{avg,hkl}) = N_{hkl} \left(\sum_L N_{L,atoms} \right) + 2 N_{hkl} \left(\sum_L N_{L,atoms} \right) + N_L (l_{max} + N_{hkl}) \quad (12)$$

$$= 54881 * 44 + 2 * 44 * 54881 + 6 * (4915 + 54881) = 7.60 \times 10^6$$

$$O(Peaks) = N_{pks_buf} N_{ops_per_pk} + N_{pks_summed} N_{pk_shape_sum}$$

$$= 1348 * 2000 + 11000 * 150 = 4.34 \times 10^6$$

$N_{hkl} \left(\sum_L N_{L,atoms} \right)$ corresponds to $\sum_L Layer_{L,hkl}$ in Eq. 2 which relates to the calculation of the structure factors for the six different layers. l_{max} corresponds to the maximum l value encountered in the calculation of $V_{L,hkl}$ of Eq. 3.

For hkl dependent peak broadening traditional Rietveld programs, for example TOPAS Version 5 (Coelho, 2015) and earlier, would require the calculation of $N_{str} N_{hkl} = 2.5 \times 10^8$ peaks. TOPAS Version 6 uses a peaks buffer that is hkl dependent (Ainsworth *et al.*, 2016) to greatly reduce the number of peaks calculated. In this example it generates $N_{pks_buf} = 3474$ peaks which is used by the N_{str} structures. Summing the peaks from the peaks buffer to form the calculated pattern is also a process that can be streamlined such that peaks of a similar shape that lie in between adjacent data points have their intensities appropriated to the adjacent data points. In this manner the number of peaks summed onto the calculated pattern

is typically less than the number of data points in the pattern; in the present example $N_{pks_summed} \cong 11000$ as opposed to the number of reflections which is 54881.

Both Eq. 11 and Eq. 12 are approximate in regards to computing time as each operation can amount to 4 to 10 floating point operations with the actual time taken being hardware dependent; they do, however, give an idea of the magnitude of the computation, especially when comparing the values between Eq. 11 and Eq. 12. For derivatives the computational effort is dependent on which parameters are refined. The number of operations for calculating a peak shape derivative scales by $O(Peaks)$; for structural parameters it scales by N_{hkl} , see Eq. 9, and $O(N_{pks_summed} N_{pk_shape_sum})$. Thus even though $O(N_{pks_summed} N_{pk_shape_sum})$ is relatively small compared to $I_{avg,hkl}$ of Eq. 12 it can become significant when many structural parameters are refined.

The resulting computational speedup of Eq. 12 compared to traditional Rietveld refinement (Eq. 11) is in the order of 10^4 ; this would further increase for structures with more atoms in each layer, or, for cases where more complex peak shapes are used.

8. Test data

Figure 1 shows the current implementation fitting data generated with the test input file dia.dat for stacking-faulted diamond distributed with the DIFFaX suite (Treacy, 1991) with a 0.7 probability (pa) for $N_v=200$ and for the cases of $N_{str}=1$ and $N_{str}=200$. The much improved fit for $N_{str}=200$ (Figure 1b), $R_{wp}=0.58\%$ as opposed to $R_{wp}=3.76\%$, is due to the averaging procedure of Eq. 4 and the smoothing function of Eq. 6. No weighting was given to the data during refinement and the formula used to calculate R_{wp} is shown in Eq. 13, where Y_o and Y_c correspond to observed (here DIFFaX-simulated) and calculated intensities.

$$R_{wp} = 100 \sqrt{\frac{\sum_{m=1}^{N_{data}} (Y_{o,m} - Y_{c,m})^2}{\sum_{m=1}^{N_{data}} (Y_{o,m})^2}} \quad (13)$$

The time taken to perform the first 8 iterations for $N_{str}=1$ is 1.92 seconds; for $N_{str}=200$ it is 1.99 seconds. Refinement times, however, depend on what is being refined as various calculated data is saved during refinement and calculated only once. It is therefore more relevant to consider the time taken to calculate the pattern itself; for ten pattern calculations the time for $N_{str}=1$ is 0.67 seconds; for $N_{str}=200$ it is 1.26 seconds; or around twice as slow for 200 times more structure factors calculations. Working with the same DIFFaX data of Figure 1, Ainsworth *et al.* (2016) used $N_v=5000$ stacking vectors but with $N_{str}=1$. Using the input file of Table 2 but with these parameters ($N_{hkl}=236902$, $N_{pks_buf}=2760$) results in an R_{wp} of 1.28% and a time to calculate the pattern ten times of 6.8 seconds. This demonstrates the great benefit of averaging in both computing time and accuracy.

To explore the influence of N_{str} and N_v the simulated data were initially fitted with $N_{str}=200$ and $N_v=5000$ and without the use of the smoothing function; five peak shape parameters, a zero error and one Rietveld scale parameter were initially refined. These large N_{str} and N_v values results in a low R_{wp} of 0.465% as shown in Table 3. The peak shape parameters and zero error were then fixed and the smoothing function of Eq. 6 included in refinements using different N_{str} and N_v values; Table 3 shows the results. For equivalent faulting statistics ($N_{str} \times N_v$ equal) the R_{wp} values are very similar. A reduction in R_{wp} is seen as $N_{str} \times N_v$ increases. For $N_v \geq 100$ and for $N_{str} \times N_v \geq 100000$ R_{wp} values are similar. The case of $N_{str}=200$, $N_v=200$ has a marginally higher $R_{wp}=0.509\%$, but still with the excellent fit seen in Fig. 1b. These data demonstrate that small supercells (low N_v) can approximate much larger supercells when averaging and the smoothing function of Eq. 6 are used.

To further test the I_{avg} procedure, the test structure described in Tables 4 and 5 was used to generate a complex test pattern with $pa=0.7$ and with two different layer heights of 5 and 6 Å ($\varepsilon=1$ in Table 5) with the other five lattice parameters being $a=3$ Å, $b=4$ Å, $\alpha=90^\circ$, $\beta=90^\circ$ and $\gamma=95^\circ$. For accuracy a small step size of $0.01^\circ 2\theta$ was used and the peak shape comprised a Gaussian instrument function with a FWHM of twice the step size; this results in a pattern that is essentially the aberration function produced from stacking faults. The pattern was generated using a normal Rietveld calculation averaging 200 individual structures each with a large supercell comprising $N_v=2000$ layers and each with separate c lattice parameters corresponding to the overall heights of the stacked layers. Each unit cell therefore comprised $2000 \times 4 = 8000$ atoms for a total of 1,600,000 atomic sites in the calculation. The calculation was performed out to $40^\circ 2\theta$ which encompassed 28325 reflections. This translates to an innermost loop for the calculation of structure factors that is executed $1600000 \times 28325 = 45,176,000,000$ times; the pattern calculation time was long at 1828 seconds. Fitting to this test pattern using I_{avg} with $N_{str}=200$ and $N_v=2000$ (identical faulting statistics to the generation procedure) produces the good fit shown in Fig. 2 where $R_{wp}=3.27(20)\%$; the calculation time for the first iteration was only 1.2 seconds. The non-zero R_{wp} is due to the faulting statistics represented by the transition probability matrix and remaining minor ripples in the (instrument/sample unbroadened) test pattern. Setting $N_v=400$ and using the smoothing function of Eq. 6 results in $R_{wp}=4.08(25)\%$ and a 10-fold faster calculation of 0.12 seconds. Thus I_{avg} correctly describes the Rietveld generated pattern with two layer heights in an extremely fast manner when a relatively small N_v is used together with the smoothing function.

To further show the benefits of the smoothing function of Eq. 6 the same structure but with $\alpha=75^\circ$, $\beta=85^\circ$ and $\gamma=95^\circ$ was used to generate a test pattern out to $150^\circ 2\theta$; no smoothing function was used and the pattern was again generated using I_{avg} with $N_{str}=200$ and $N_v=2000$.

Fitting to this test pattern with the same faulting statistics (equal $N_{str} \times N_v$) we put $N_{str}=2000$ and $N_v=200$. Fig. 3a shows the poor fit when the smoothing function is not used. Fig. 3b show the fit when the smoothing function is used resulting in $R_{wp}=1.77(13)\%$ out to $150^\circ 2\theta$. Setting $N_{str}=200$ gives an $R_{wp}=1.82(5)\%$. Thus the smoothing function with a small supercell accurately describes much larger supercells for the complex case of varying layer heights.

Fig. 4 show the fit to a simulated pattern of the $\text{La}_2\text{O}_2\text{Cu}_{0.667}\text{Cd}_{0.667}\text{Se}_2$ structure reported by Ainsworth *et al.* (2016) which comprised three layer types. The simulated pattern used a step size of $0.005^\circ 2\theta$, peak shapes set to a Gaussian with a FWHM of $0.01^\circ 2\theta$, $N_{str}=200$ and $N_v=2000$ and no smoothing function. The calculated pattern for $N_{str}=2000$ and $N_v=200$ with the smoothing function resulted in $R_{wp}=1.335(39)\%$; setting $N_{str}=200$ and $N_v=200$ gives $R_{wp}=1.63(27)\%$. These R_{wp} values can be compared to 1.45% for $N_{str}=200$ and $N_v=2000$ (same values as those used for the simulated pattern; non-zero R_{wp} due to the faulting statistics). Fig. 5 shows R_{wp} as a function of iteration time from Rietveld refinement using I_{avg} , the smoothing function, $N_{str}=200$ and $N_v=200$ for the $\text{La}_2\text{O}_2\text{Cu}_{0.667}\text{Cd}_{0.667}\text{Se}_2$ test pattern. The refinement range of was set to 1.5 to $30^\circ 2\theta$ resulting in 41272 reflections. The refinement comprised 15 site coordinate parameters, 3 layer height parameters, one lattice parameter, one Rietveld scale parameter and τ from the smoothing function of Eq. 6b. Layer height parameters were initially set to $\sim 0.05 \text{ \AA}$ from their optimal values. Fig. 5 show the fast convergence of these stacking vector parameters. The five seconds to refinement convergence is significant as it allows for hands on real time analysis of such data sets.

These test data suggest that large N_v supercells are not necessary to fit complex patterns. In addition, real data has considerable intrinsic broadening due to instrument, emission profile and specimen type aberrations such as crystallite size and strain broadening. Thus even though the smoothing function of Eq. 6 is important in allowing small- N_v fits to simulated test data with narrow intrinsic peaks, the need with real data is lower.

9. Real data Analysis

We can further explore the speed and accuracy gains enabled by these developments using one of the data sets described in Ainsworth et al. (2016). We choose the $\text{La}_2\text{O}_2\text{Cu}_{0.667}\text{Cd}_{0.667}\text{Se}_2$ composition discussed above which is a layered oxychalcogenide that can be described in terms of alternating fluorite-like $[\text{La}_2\text{O}_2]^{2+}$ slabs and either $[\text{CdSe}_2]^{2-}$ or $[\text{Cu}_2\text{Se}_2]^{2-}$ anti-fluorite-derived slabs. The ideal structure has a 2:1 ordered sequence of metal chalcogenide layers ($\text{Cd1}|\underline{\text{Cd2}}|\text{Cu3}|\underline{\text{Cd4}}|\text{Cd5}|\text{Cu6}$), where each underlined Cd layer is offset by $(a/2, b/2)$. Two types of faults were investigated: a *pa* type fault involving an additional layer of either metal to form local $(\text{Cu}|\text{Cu}|\text{Cd}|\text{Cd})$ or $(\text{Cu}|\text{Cd}|\text{Cd}|\text{Cd})$ sequences; and a *pb* fault involving Cd layers being offset by $(a/2, b/2)$ either side of a Cu layer. In the original paper the dependence of R_{wp} on fault probabilities *pa* and *pb* was investigated using a model with six $[\text{LaOLaSe}(\text{Cu}_2/\text{Cd})\text{Se}]$ layers (three unique) stacked to form a supercell with 960 layers in total ($N_{\text{str}}=1, N_L=6, N_v=960$).

Firstly, we investigate the stacking generation method as a function of the constant q described in step (2) of Table 1. Figure 6a shows the frequency distribution of R_{wp} for 500 Rietveld refinements for the cases of $q=0.5$ and $q=\infty$; in the latter case there is no constraint on the number of faults generated in the structures and the average R_{wp} is slightly larger than the value for $q=0.5$. Figure 6b shows corresponding generated *pa* values pa_{gen} ; a much narrower spread in pa_{gen} is found for $q=0.5$ as expected. For this reason $q=0.5$ is used in the analysis below.

Table 6 shows the influence of different values for N_{str} and N_v on R_{wp} and pa_{gen} using our averaging approach. Figure 7 shows the influence of different N_{str} values for the R_{wp} surfaces obtained for $N_v=960$ with *pa* and *pb* values between 0.0 and 0.1 with different procedures. Figure 7a show the surface of minimum R_{wp} values obtained from 100 $N_{\text{str}}=1$ trial structures for each of 64 *pa/pb* combinations as reported in the original paper. The entire surface took

around 15 hours to calculate on an i7 3.4 GHz single core desktop computer. Figure 7b shows the equivalent R_{wp} surface determined using our averaging processes with $N_{str}=100$, which took 160 s on the same computer. The speed gain is a combination of the smaller number of refinements required (64 vs 6400), the use of a more efficient algorithm to create constrained stacking sequences, and other improvements to TOPAS. A more direct comparison of the speed gain from averaging can be obtained by comparing the time to do 64×100 refinements with $N_{str}=1$ (~8000 s) with 64 $N_{str}=100$ refinements (160 s). A speed gain of around 50 fold is observed as expected from the discussion above. Using the convolution approach of Section 5 an equivalent surface can be calculated for $N_v=96$ in around 20 s.

The averaging process with $N_{str}=100$ also leads to a much better defined minimum in the R_{wp} surface as shown by the contour levels and minimum–maximum R_{wp} ranges in Figures 7a and 7b (2.80–7.02% and 2.71–8.59% respectively). The flatter surface in Figure 7a arises from the range of R_{wp} values encountered for different individual $N_{str}=1$ sequences. Figure 7c shows this effect by superimposing the single R_{wp} value from the $N_{str}=100$ refinement (large points and surface mesh) with the 100 R_{wp} values from $N_{str}=1$ refinements (small points). For each set pa/pb combination we see a vertical “rod” of R_{wp} values reflecting the random nature of the individual stacking sequences generated in each structure. Close to the overall R_{wp} minimum the $N_{str}=100$ refinements lie at the minimum of these rods. For larger pa/pb values some $N_{str}=1$ refinements show lower R_{wp} , but this is due to individual structures having fewer faults than the expectation value. Figure 7d shows an equivalent plot to Figure 7c, but with R_{wp} values for the $N_{str}=1$ refinements plotted against the actual number of faults present in each stack. On this plot $N_{str}=100$ models lie essentially on the minimum- R_{wp} surface. This plot also shows that $N_{str}=1$ R_{wp} values have significant scatter; around the R_{wp} minimum individual R_{wp} values can be up to 1% higher than the lowest value.

As expected from the smooth R_{wp} surfaces in Figure 7, it is relatively straightforward to directly optimise values of pa and pb . There are a number of ways of doing this based on Monte-Carlo or simulated annealing type approaches (e.g. Bette 2015). Here a straightforward downhill or greedy algorithm is effective. Figure 8 shows typical minimisation pathways for this problem. Each minimisation was complete in a few minutes of computer time.

Conclusion

In conclusion we show how the use of efficient algorithms for calculating structure factors, efficient pattern averaging, speed-ups in powder pattern calculation using a peaks buffer, a smoothing procedure that allows for the use of much smaller supercells and efficient stacking algorithms lead to massive speed gains in the Rietveld analysis of stacking-faulted structures. Real speed gains of the order of 10^4 have been obtained, allowing rapid combined structural and microstructural analysis of complex systems. These processes have been implemented in a general way in the computer program TOPAS Version 6, and we believe they will have wide applicability.

References

- Ainsworth, C. M.; Lewis, J. W.; Wang, C.; Johnstone, H. E.; Mendhis, B. G.; Brand, H. E. A.; Coelho, A. A.; Evans, J. S. O. (2016). *Chem. Mater.*, **28**, 3184–3195.
- Bette, S.; Dinnebier, R. E.; Freyer, D. (2015). *J. Appl. Cryst.* **48**, 1706-171.
- Bruker (2009). DIFFRACplus TOPAS4.2. *Technical Reference Manual*. Bruker AXS, Karlsruhe, Germany.
- Bruker AXS (2016). TOPAS, V6.0. Bruker AXS, Karlsruhe, Germany
- Cheary, R. W. & Coelho, A. (1992). *J. Appl. Cryst.* **25**, 109-121.
- Coelho, A. A. *TOPAS Academic: General Profile and Structure Analysis Software for Powder Diffraction Data*; 5th ed.; Bruker AXS: Karlsruhe, Germany, 2012.
- Coelho, A. A. (2015). TOPAS-Academic V5. <http://www.topas-academic.net/>
- Lutterotti, L., Voltolini, M., Wenk, H. R., Bandyopadhyay, K., & Vanorio, T. (2010). *American Mineralogist*, **95**(1), 98-103.
- Rietveld, H.M. (1969): A profile refinement method for nuclear and magnetic structures. *J. Appl. Cryst.*, **2**, 65-71.
- Treacy, M. M. J.; Newsam, J. M.; Deem, M. W. (1991) *Proceedings of the Royal Society-Mathematical and Physical Sciences*, **433**, 499.
- Ufer, K., Roth, G., Kleeberg, R., Stanjek, H., Dohrmann, R. & Bergmann, J. (2004). *Z. Kristallogr.* **219**, 519–527.
- Wang, X.; Hart, R.D.; Li, J.; McDonald, R. G.; van Riessen, A. (2012). *J. Appl. Cryst.* **45**, 1295–1302.
- Young, R.A. (1993). The Rietveld Method, edited by R.A. Young, IUCr Book Series, Oxford University Press 1993, 1-39. “*Introduction to the Rietveld method*” .

- 1) For each structure generate N_v layers based on the probability matrix and form a single summation over all structures to produce the summation matrix \mathbf{S}_{gen} of layer i to j transitions.
- 2) Terminate the generation of \mathbf{S}_{gen} when the following is obeyed for each of its elements:

$$|\mathbf{S}_{pred,i,j} - \mathbf{S}_{gen,i,j}| < q \sqrt{\mathbf{S}_{pred,i,j}}$$

where the subscripts i and j are over the elements of \mathbf{S} and $q = 0.5$.

- 3) Chose one of the generated structures at random.
- 4) Add a layer to the randomly chosen structure and keep the last N_v layers.
- 5) Go to step (2).

Table 1 Steps used in the generation of stacking sequences for all structures.

```

xdd diffax_dia.xye
  LP_Factor(0) bkg 1
  start_X 10 finish_X 149.6
  rebin_with_dx_of 0.01
  lam ymin_on_ymax 0.0001 la 1 lo 1.5405754 lh 1e-5
  Zero_Error(@, 0.0025)
  weighting 1
  str
    space_group P1
    a 2.518156 b 2.518156 c = Get(generated_c); ga 120
    prm !pa 0.7 prm !h 2.05870
    prm s 1 min 1e-15 scale = s 1e-6 / (Nv Nstr);
    generate_stack_sequences {
      number_of_sequences Nstr 200
      number_of_stacks_per_sequence Nv 5000
      Transition(A, h)
        to A = pa; a_add = 2/3; b_add = 1/3;
        to B = 1-pa; a_add = 0; b_add = 0;
      Transition(B, h)
        to A = 1-pa; a_add = 0; b_add = 0;
        to B = pa; a_add = -2/3; b_add = -1/3;
    }
    site C1 x = -1/3; y = -1/6; z = -0.125/Nv; occ C 1 beq 1 layer A
    site C2 x = 1/3; y = 1/6; z = 0.125/Nv; occ C 1 beq 1 layer A
    site C3 x = 1/3; y = 1/6; z = -0.125/Nv; occ C 1 beq 1 layer B
    site C4 x = -1/3; y = -1/6; z = 0.125/Nv; occ C 1 beq 1 layer B
    peak_buffer_based_on = Xo; peak_buffer_based_on_tol 0.1
    TCHZ_Peak_Type(@,0.01,@,-0.01,@,0.01,,0,@,0.1,@,0.01) 'Instrument function

```

Table 2 Input file used for fitting to DIFFaX diamond data for $N_{str}=200$ and $N_v=5000$.

N_{str}	1	1	200	1000	500	200	10000	5000	200
N_v	200	5000	200	100	200	500	100	200	5000
$N_{str} \times N_v$	200	5000	40000	100000			1000000		
N_{hkl}	9474	236902	9474	2361	9474	23611	4724	9474	236902
N_{pks_buf}	2532	2760	2532	1122	2532	5305	1122	2532	2760
Avg. R_{wp} (%)	2.90	1.28	0.509	0.498	0.489	0.483	0.471	0.458	0.465
Std Dev. R_{wp}	0.53	0.17	0.040	0.020	0.025	0.027	0.007	0.006	0.006

Table 3 Average and standard deviation of R_{wp} over 30 refinements each on DIFFaX-simulated diamond data for $pa=0.7$ for various N_{str} and N_v values. Refinement range 10 to $150^\circ 2\theta$. The smoothing function was applied to all refinements except for the cases with $N_v=5000$.

Layer	Atom	x	y	z	Beq (\AA^2)
1	C	$-1/3$	$-1/6$	-0.125	1
1	C	$1/3$	$1/6$	0.125	1
2	C	$1/3$	$1/6$	-0.125	1
2	C	$-1/3$	$-1/6$	0.125	1

Table 4 Fractional atomic coordinates used to generate test patterns. All sites occupied with carbon atoms with an occupancy of 1.

From layer 1

To layer 1 with a probability of pa and shifted by the vector $\{2/3, 1/3, c/N_v\}$

To layer 2 with a probability of $1-pa$ and shifted by the vector $\{0, 0, c/N_v\}$

From layer 2

To layer 1 with a probability of $1-pa$ and shifted by the vector $\{2/3, 0, (c+\varepsilon)/N_v\}$

To layer 2 with a probability of pa and shifted by the vector $\{2/3, 2/3, (c+\varepsilon)/N_v\}$

Table 5 Transition matrix and stacking vectors used to generate test patterns. Shifts in ab plane expressed in fractional coordinates; those along stacking direction in Å.

N_{str}	1	100	200	1	25	100	200
N_v	960	960	960	192	192	192	192
N_{hkl}	54877	54877	54877	10973	10973	10973	10973
N_{pks_buf}	3474	3474	3474	1326	1326	1326	1326
Average $R_{wp}(\%)$	3.014	2.933	2.933	3.591	2.780	2.751	2.748
Std Dev. $R_{wp}(\%)$	0.146	0.010	0.008	0.662	0.044	0.019	0.014
pa_{gen}	0.0096	0.01103	0.01101	0.009	0.0109	0.01101	0.01101
Std Dev. pa_{gen}	0.0014	0.00012	0.00008	0.006	0.0006	0.00025	0.00022
Time (s)	0.71	1.33	1.96	0.45	0.49	0.56	0.66

Table 6 Average and standard deviation of R_{wp} and pa_{gen} over 500 refinements for $\text{La}_2\text{O}_2\text{Cu}_{0.667}\text{Cd}_{0.667}\text{Se}_2$ (data from Ainsworth *et al.*, 2016) for various N_{str} and N_v values. Refinement range 1.5 to $20^\circ 2\theta$. The approximate relative time required for each of the 500 refinements is given.

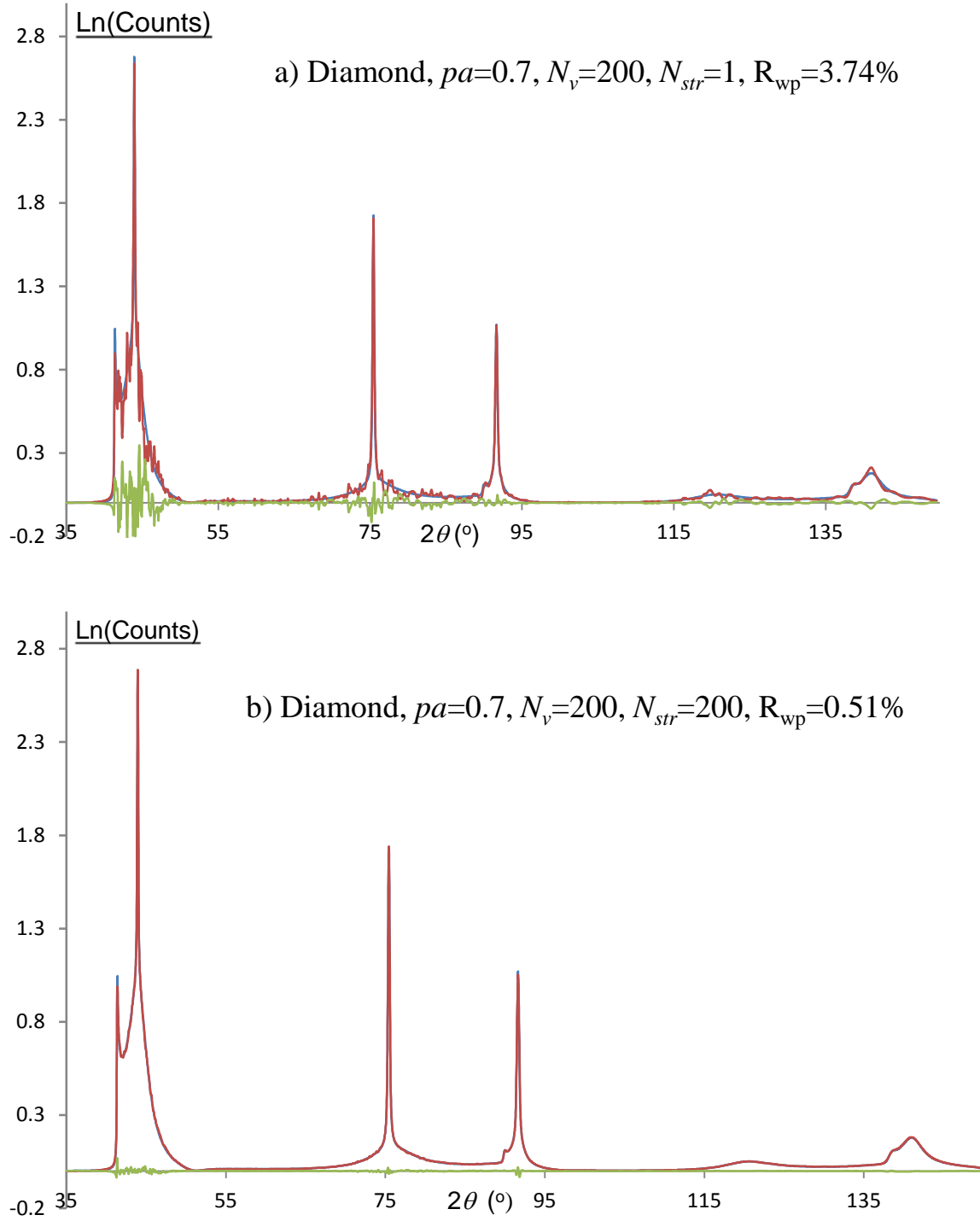


Figure 1 Fit to simulated diamond data from DIFFaX for the cases of a) $N_{str}=1$ and b) $N_{str}=200$. In both cases $N_{hkl}=9474$ and $N_{pks_buf}=2532$ peaks. Red, blue and grey lines correspond to calculated, DIFFaX simulation and difference plots.

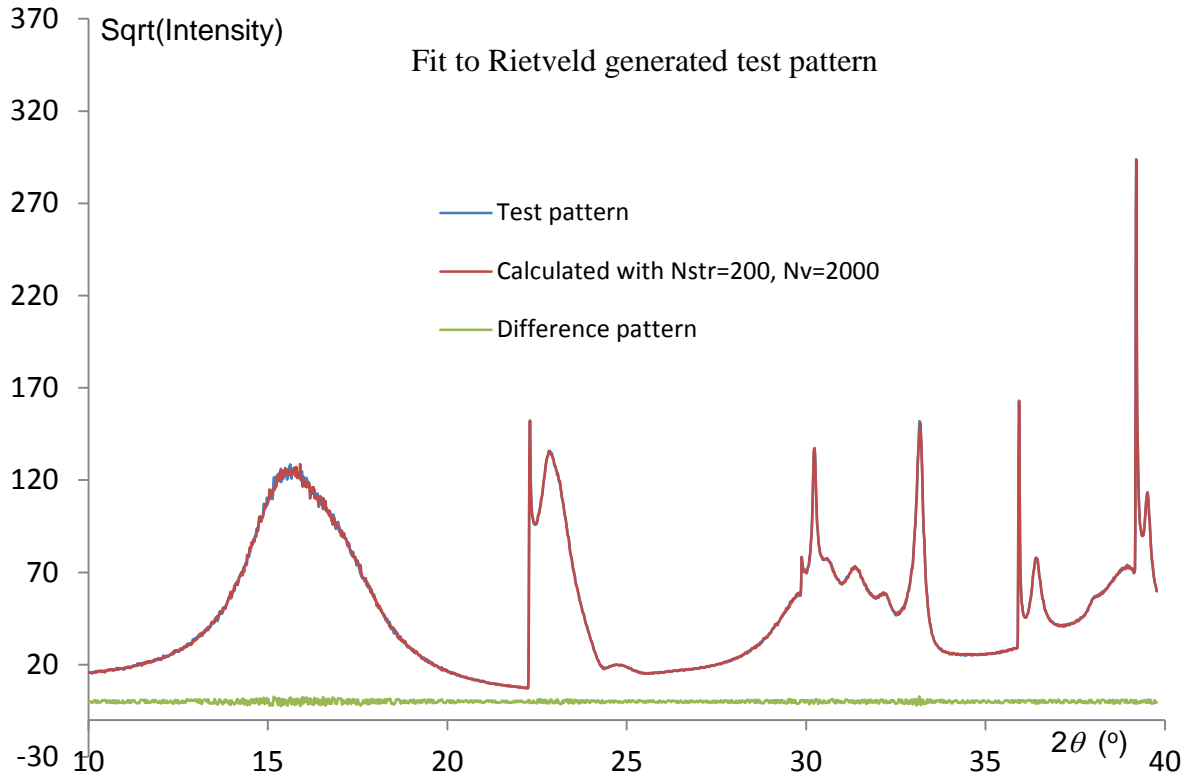


Figure 2 Fit to Rietveld-generated pattern comprising 200 structures each with 2000 layers and each with its own c lattice parameter determined from the different layer thicknesses and faulting. The calculated pattern used the I_{avg} implementation with a single c lattice parameter.

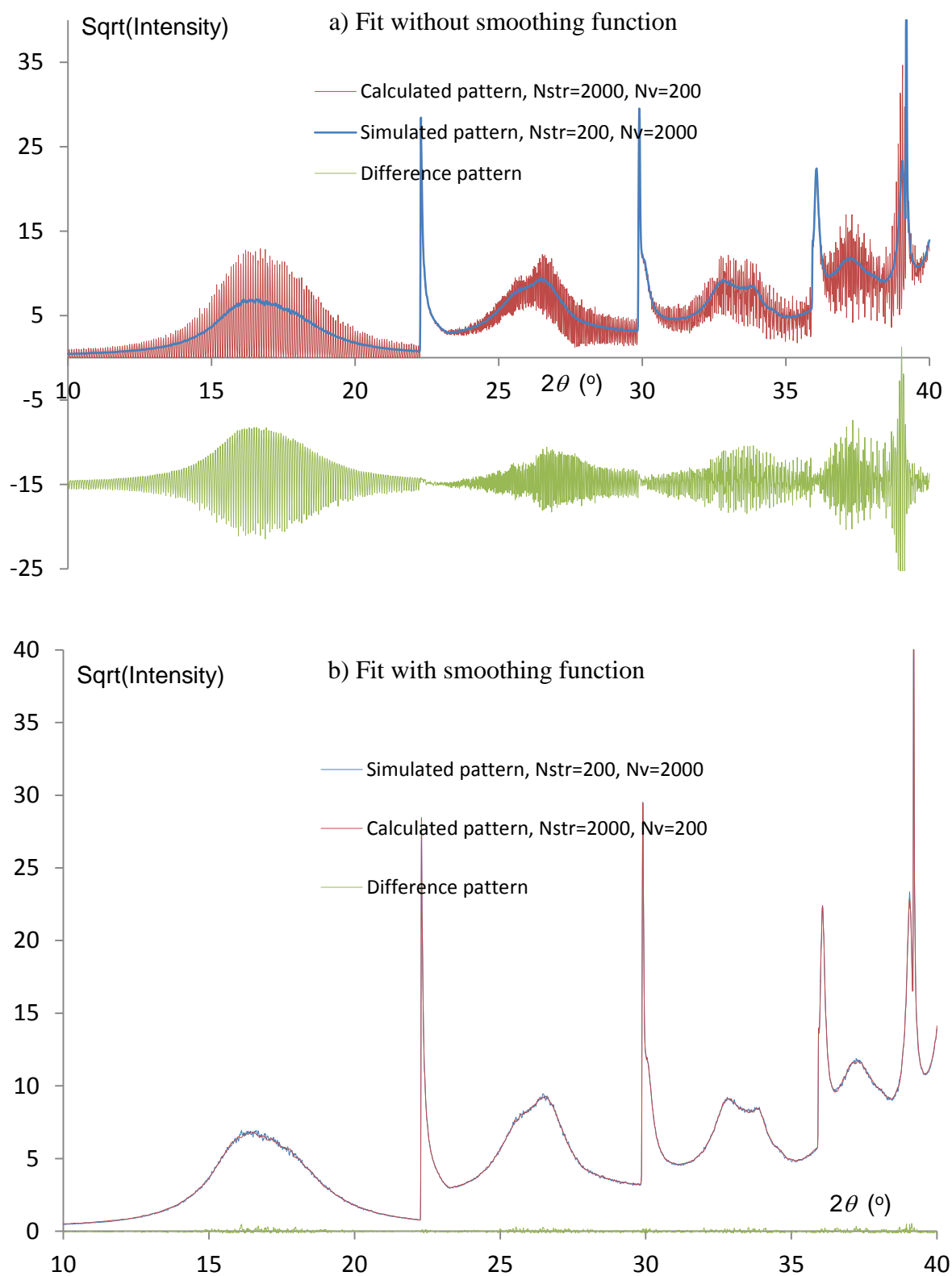


Figure 3 Fit to a simulated test pattern created with $N_v=2000$ and no smoothing function using a calculated pattern created with $N_v=200$ for the cases a) no smoothing function and b) with the smoothing function.

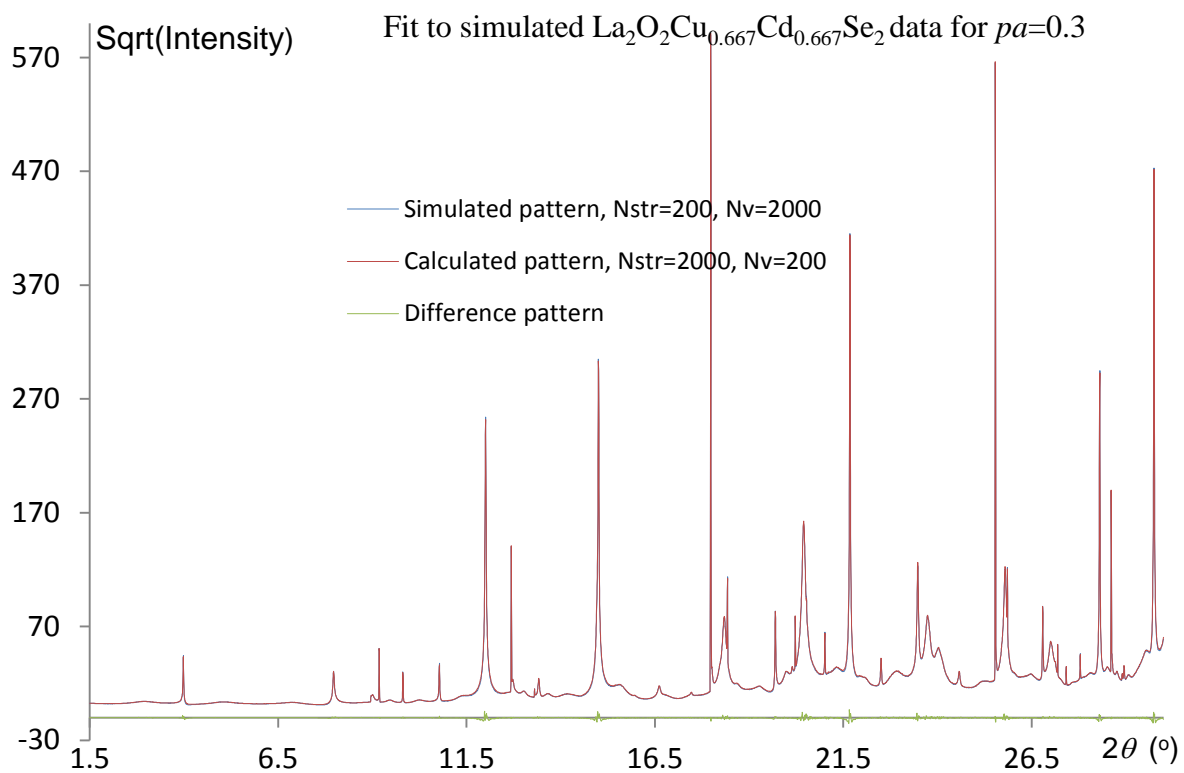


Figure 4 Fit to a simulated $\text{La}_2\text{O}_2\text{Cu}_{0.667}\text{Cd}_{0.667}\text{Se}_2$ test pattern created with $pa=0.3$, $N_v=2000$ and no smoothing function using a calculated pattern with $N_v=200$ and with the smoothing function.

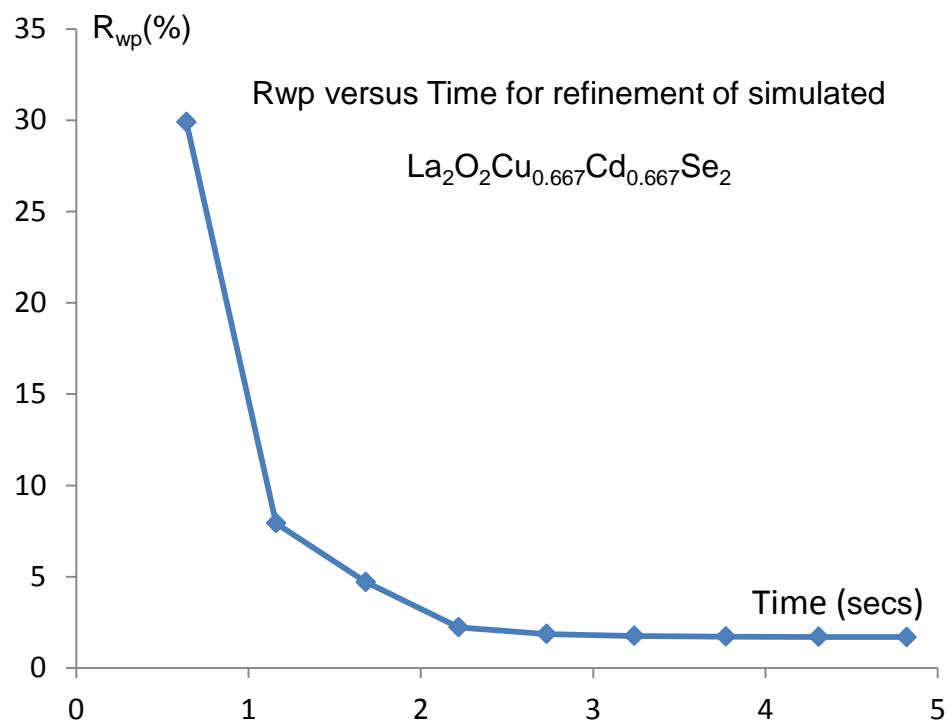


Figure 5 R_{wp} versus time from Rietveld refinement of a test $\text{La}_2\text{O}_2\text{Cu}_{0.667}\text{Cd}_{0.667}\text{Se}_2$ pattern. The refinement used $N_{str}=200$ and $N_v=200$.

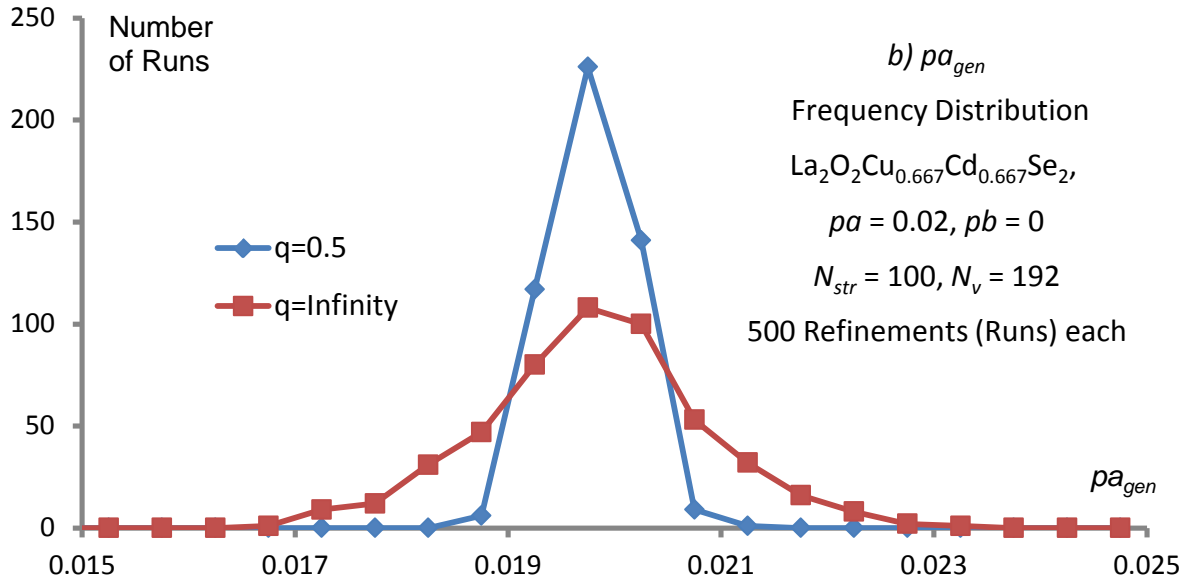
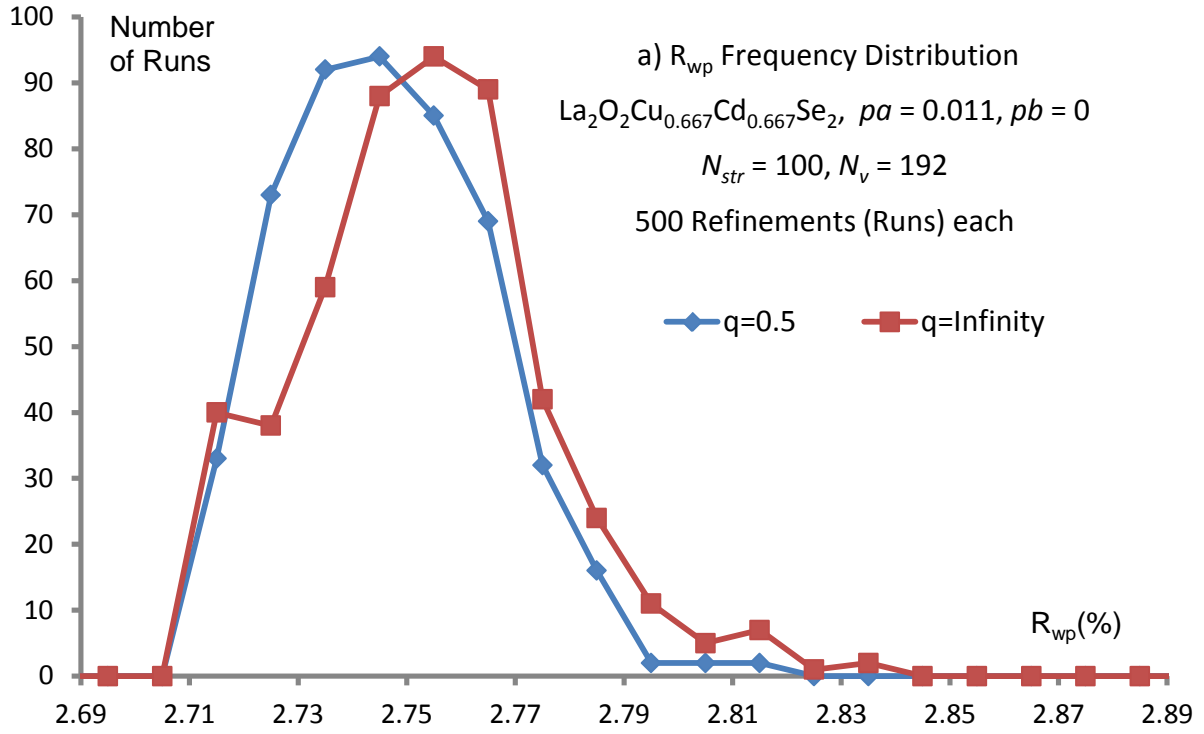


Figure 6 Frequency distribution from two sets of refinements, $q=0.5$ and $q=\infty$, each comprising 500 refinements of $\text{La}_2\text{O}_2\text{Cu}_{0.667}\text{Cd}_{0.667}\text{Se}_2$ with $N_{str}=100$ and $N_v=192$ for a) R_{wp} for b) pa_{gen} .

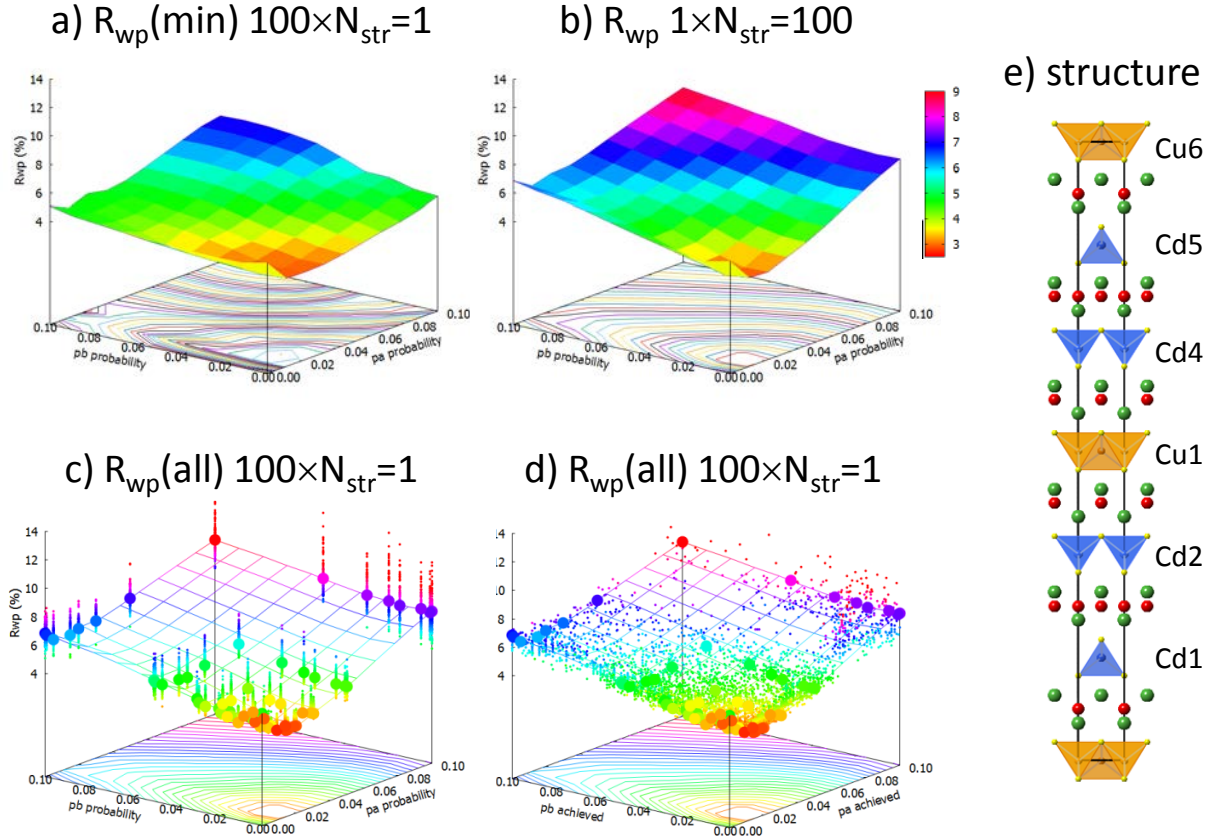


Figure 7 3D plots showing Rwp values for refinements of different $\text{La}_2\text{O}_2\text{Cu}_{0.667}\text{Cd}_{0.667}\text{Se}_2$ stacking models. (a) shows the minimum Rwp value from 100 repeat refinements of models with $N_v = 960$ and $N_{str} = 1$ and different pa/pb values. (b) shows the Rwp value from single models with $N_v = 960$ and $N_{str} = 100$. (c) plots the 100 Rwp values for $N_{str} = 1$ (small points) and the single value for $N_{str} = 100$ (large points). (d) is as (c) but with data plotted against the actual number of faults generated in each sequence. The same z-range and colouring is used in each plot. Lines in (c) and (d) are to guide the eye. (e) shows the structure in terms of $[\text{La}_2\text{O}_2]^{2+}$ (green/red), $[\text{CdSe}_2]^{2-}$ (blue/yellow) and $[\text{Cu}_2\text{Se}_2]^{2-}$ (brown/yellow) slabs.

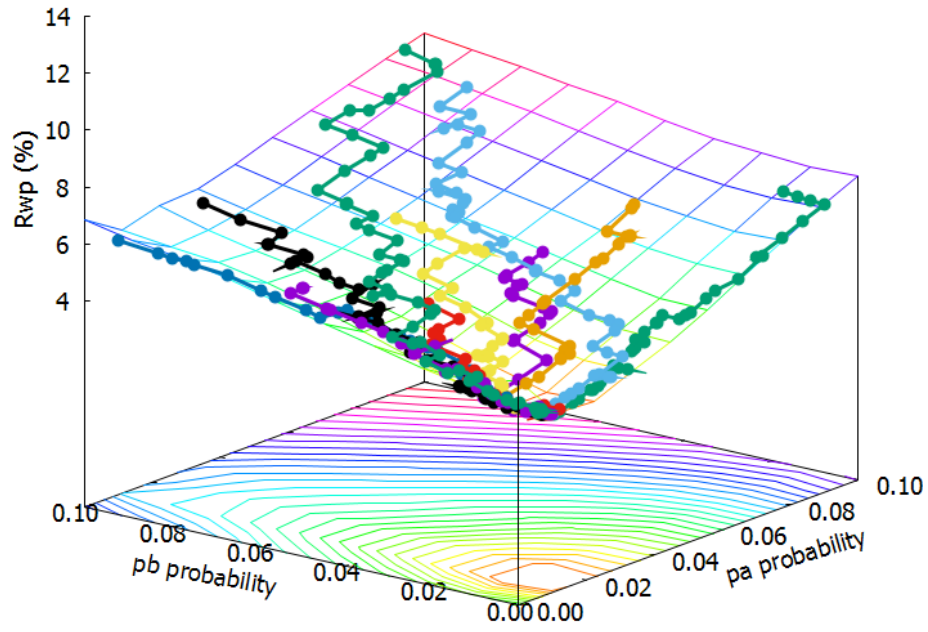


Figure 8 Minimisation pathway using simple downhill algorithm. Minimisation pathways from ten different random starting points shown on an equivalent surface to that of Figure 2b.